1 Data Acquisition Electronics and Programmer's Model

DRAFT TN38
October 26, 2001
V. Sandberg

The QT electronics provides the front-end PMT pulse processing that creates two analog outputs, one proportional to the integrated charge in the pulse as a function of time and the second a linear voltage ramp that begins at a fixed threshold (corresponding to between 1/5 and 1/4 of a photoelectron, the variations due to the gain mismatches amongst the tubes) on the leading edge of the PMT pulse. The time series illustrating these waveforms is shown in Fig. **??**, the front-end circuit schematic is shown in Fig. 1, an updated front-end circuit is shown in Fig. 2, and the schematic of the digital and VME circuitry is shown in Fig. 3. The FADC samples at 10 MHz to 8-bit accuracy. In the case of the time voltage ramp the eight bits resolves the analog signal into 255 channels and allowing for 5 channels of offset from zero and full scale gives a time resolution of 0.8ns. In the case of the charge, a calibration of 8 ADC channels equals 1 photoelectron of charge constrains the dynamic range to be approximately 30 photoelectrons full scale.

The front-end encodes the data of interest to the experiment; namely, charge and fine time information, in analog voltages that are accurately digitized by the FADC. The system does not attempt to record the PMT waveform. A transient digitization would require a larger sampling rate to meet the Nyquist criteria for good pulse fidelity, and would present an enormous computational burden on the downstream DAQ computers. Our more modest approach gives us the charge ("Q") and time ("T") of a series of arriving PMT pulses in a time series spaced at 100 ns intervals.

Any PMT pulse arriving is terminated in 50 ohms and buffered by an emitter follower. The input to any channel may be observed through the monitor which is connected to the input through a series 450 ohm resistor. This port may also be used to inject a test pulse to test the particular channel of QT electronics. The buffered signal is sent to the charge integrator and the discriminator.

The integrator circuit is a common base stage. The PMT signal is applied to the gain determining resistor connected to the emitter of the 2N5770 transistor. The PMT pulse current (equal to the PMT signal voltage divided by the value of the gain determining resistor, as the transistor's emitter may be considered as a virtual ground to this approximation) is transfered by the transistor to its collector circuit. The collector acts as a current source driving charge onto the 100 pF integrating capacitor (a silver mica low leakage, low inductance capacitor). The voltage across the capacitor (called Vq in Fig. **??**) is proportional

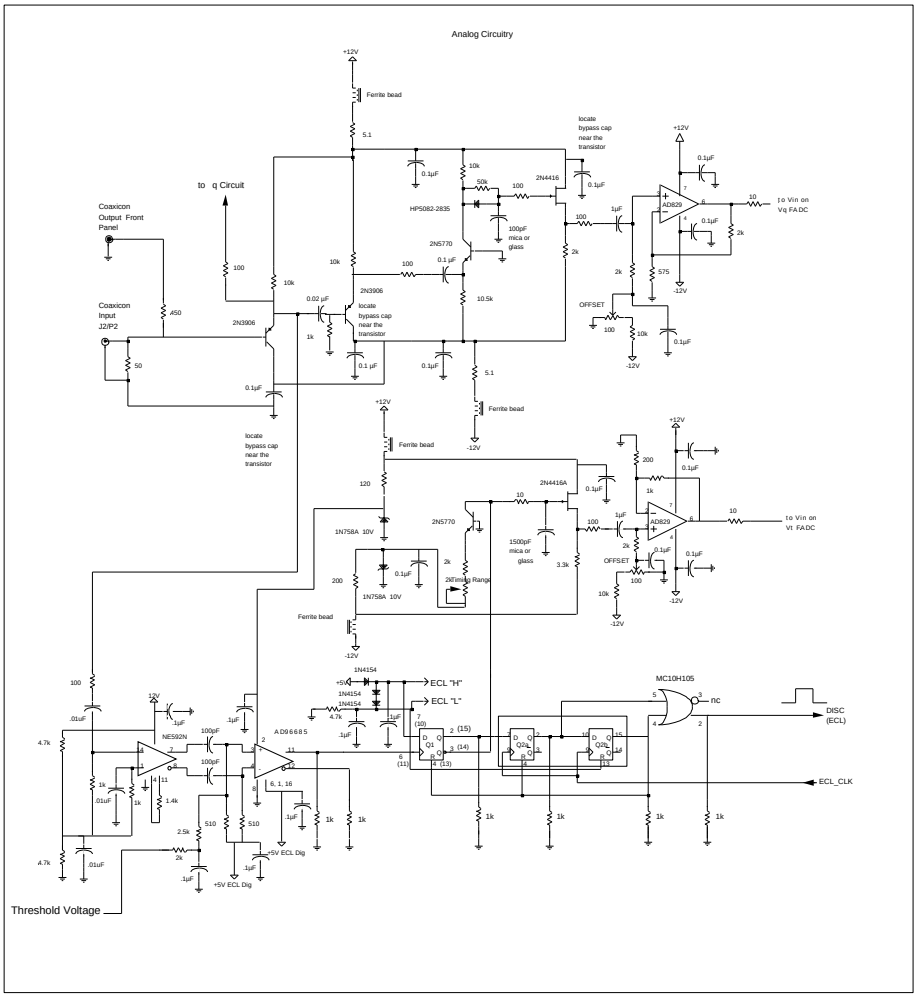

Figure 1: The front-end analog electronics for each PMT channel. The upper part of the schematic is the charge integrating circuit. It consists of a common base stage operating as a fast current sink connected to the integrating capacitor. The lower part shows the time interpolating circuit that starts a ramp on the leading edge of the discriminator pulse and resets the ramp on the second subsequent clock pulse. The discriminator threshold is typically set to fire for an input PMT pulse of -7mV.

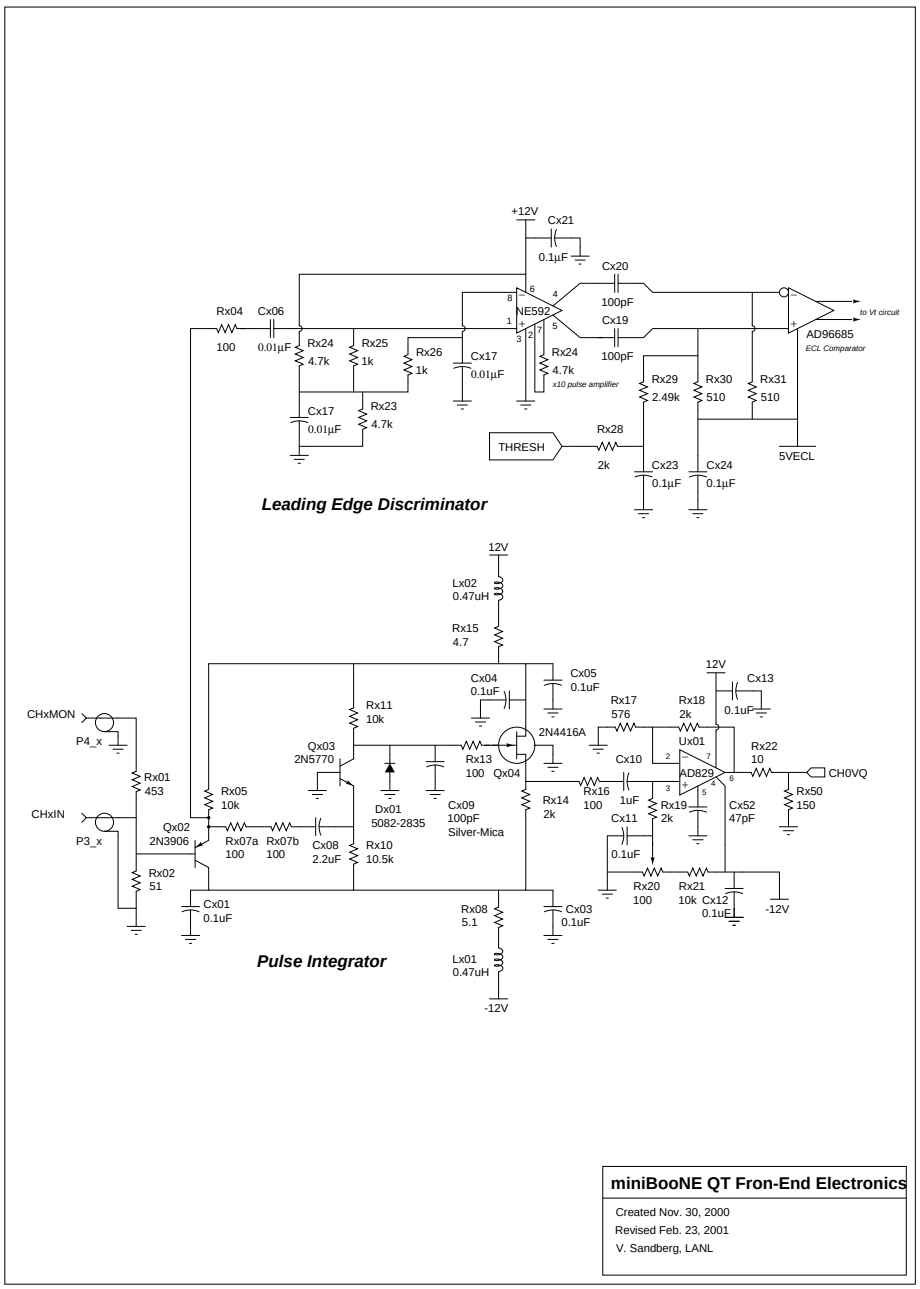

Figure 2: Updated front-end analog electronics for each PMT channel.

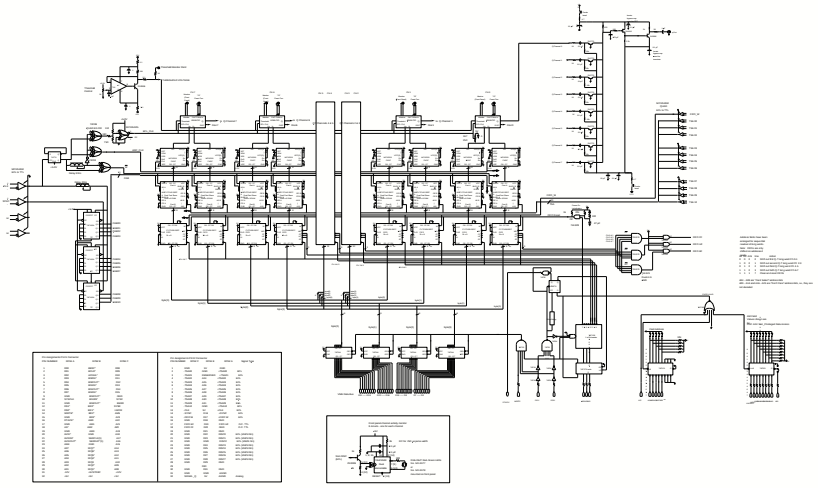

Figure 3: The front-end digital electronics on a QT card. The FADCs feed the dual-ported memories, which in turn feed the FIFOs with selected time-stamp data. Shown at the bottom of the figure is the VME addres decoding and bus interface logic.

to the charge in the PMT pulse. It is buffered by the 2N4416 FET (configured as a source follower). The FET's source voltage is amplified by an AD829 low noise bipolar op amp, which in turn drives the MC10319 FADC. (Note: These were the best parts available when this system was designed in 1992. Today, in 1999, we would take advantage of subsequent developments and realize improved performance with a smaller parts count. The system described here was built to meet the requirements of good speed and low noise for reasonable costs.) The integrating capacitor is in shunt with the 10k ohm collector resistor, which creates an RC circuit with a time constant of 1 μ sec. Properly speaking, the PMT pulse in convoluted in time with the RC. For sample times of order 100 ns the voltage is close to the integral of the pulse current. For longer times the capacitor discharges and re-zeros its baseline. In this way the digitized time series of Vq may be read to extract the charge as it flows onto the capacitor without having to gate or reset the integrator.

The discriminator consists of a video amplifier (NE592), a passive RC differenator, a fast ECL-output comparator (AD96685), a set of ECL D-flip-flops, and a stable current source. The PMT pulse is amplified and applied differentially to a fast comparator. Threshold control is provided by offsetting the comparator with a stable DC voltage, Vthreshold. If a pulse exceeds the threshold voltage the digital state is switched. The positive edge transition of this state flips the state of a D-flip-flop, which holds the state until cleared, making the discriminator blind to any subsequent comparator state changes until reset under known conditions.

A time ramp for interpolation of time between the course 100 ns clock ticks is generated from this D-flip-flop. The NPN transistor emitter follower output of the ECL flip-flop is connected to a constant current source with a value of approximately 1mA. (This value is adjustable by the calibration potentiometer and sets the full scale time range of the time-to-voltage converter.) Also connected to the output of the flip-flop is a 1500 pF capacitor. The voltage across the capacitor is normally "high", which means approximately +4.2 V. When the flip flop is triggered the state would normally go to "low", which corresponds to +3.4 V; however, the capacitor holds the voltage up and reverse biases the emitter follower inside of the integrated circuit. This decouples the emitter follower and leaves the charged capacitor to be discharged by the constant current source. The voltage across the capacitor is a linear ramp, denoted by Vt in fig. **??**. This voltage is buffered by a second op amp and digitized by a second FADC. The slope and offset are set such that an interval of two clock intervals (200 ns) corresponds to the 2 V dynamic range of the FADC.

The Vt ramp is started by the PMT pulse and it is stopped just after the second clock tick after the PMT pulse. The FADC samples the Vt at the clock ticks and records the time the pulse arrives with respect to the clock. The two tick window allows the ramp to settle. After the second tick following the pulse, the counter formed from two additional d flip-flops (which are clocked synchronously from the system clock) generates a clear and reset pulse, which resets the first flip-flop and enables the discriminator to accept another pulse. The follower connected to the timing capacitor goes "high" upon reset and pulls

up the voltage on the capacitor very quickly (in less than 50 ns with the values used here). The discriminator can process pulses within 300 ns of one another. (A possible upgrade to the circuit would allow this interval to be reduced to 100 ns or less if a faster clock is used. It may be used in the future MiniBooNE electronics.)

The digitized Vq and Vt data (called Q and T, respectively) are stored in the dual ported RAM (Cypress CY7C142). The write addresses are generated by a binary counter that counts the 10 MHz system clock. This clock is generated on each QT card from the clock signal distributed on the backplane as a differential PECL signal and is driven by the RCVR card. The counters are synchronized with a pulse that resets the counters to zero every 2×10^{12} counts. This also is derived from the common 10 MHz clock, distributed to the RCVR cards from the clock in the trigger crate. The on-card binary counter's 11-bit output completes a cycle every 2048 counts or every 204.8 μ sec.

The FADC-to-dualport loop runs continuously, independent of the trigger, and provides a 204.8 μs history of the PMT activity. The next loop to be described controls the transfer of data from the circular buffer to a FIFO. This loop is controlled by the broadcast card in the trigger crate. The broadcast card transmits FIFO write commands and memory addresses to the RCVR cards. The RCVR card contains a RCVR FIFO that records the time-stamp addresses and other header data and sends the write signal and addresses to the QT cards where they are connected to the dual-ported memories and coordinate the writing of the memory data to adjacent FIFOs (Cypress CY7C428). There it is held until read out by the crate's single board computer (SBC) (Motorola MVME167).

The crate SBC poles a status register in the RCVR card that registers the state of the RCVR FIFO's "empty flag". When the SBC detects that this FIFO is not empty it loops to a service routine that reads out the RCVR FIFO and the subsequent line of data in each of the QT FIFOs. The address space of the QT crate is so arranged that these addresses are sequential, allowing for efficient transfer of data organized with common time stamps. Once inside the SBC the data is processed for Q and T information, packaged into ethernet packets, and sent to the host computer.

The system is built in VME format and preserves the VME-Rev. C recommended practices and coding conventions. The boards will operate at significantly faster bus cycle times than the MVME167, which take 200ns per VME bus cycle.

2 Appendix 6C: Commands and Data Structures for the QT and Trigger Cards

This appendix contains a listing of the LSND Data Acquisition System Commands. The codes are the addresses of the various commands on the QT Cards, the RCVR Cards, the Broadcast Card, and the Trigger-Memory Card. By ei-

ther reading from or writing to these addresses the listed function or action is executed. Do not use these commands without a full understanding of what you are doing or you may cause serious harm.

These commands may be executed by hand when in a VxWorks shell and logged on to a monoboard computer. (Current usage is to call these single board computers (SBC). Motorola has coined the term "monoboard computers" (MBC) and we use this interchangeably with the SBC name.)

2.1 RCVR Card Introduction

The RCVR Card (receiver card) takes timestamps and event identification data off of the TSA bus, stacks the data on its own FIFO, and sends the TSA data to the QT Cards in the crate (i.e., it drives the local TSA backplane located on the P2/J2 connectors). For purposes of read out during data acquisition mode, only the reset command, "status" register" read command and the FIFO read command are of interest. The other commands are for testing and diagnostic purposes. (The RCVR Card was designed to supply the necessary test signals to operate a single QT Crate on it own as a test stand to debug QT Cards and Sigma PMT Cards.)

2.2 RCVR Card Operation

The RCVR Card is designed to be the first card in a package of cards consisting of: MBC (MonoBoard Computer) (card -1, if you will), RCVR Card (card 0), QT Card 1, QT Card 2, . . ., QT Card N (usually 16), and the Sigma PMT Card (which terminates the TSA backplane and sums the DISC bits into an 8-bit wide binary sum). The RCVR Card and the bundle of QT Cards are arranged to be contiguous in their address space. This allows for efficient readout and minimizes both local bus and VME bus access time through the use of several modes of the VMS-CHIP2 in the MVME167 MBC.

The VME bus is set to operate as: D32: A24: Non-Privileged Data Access. The AM fields (Address Modifier fields) on the decoder logic on the RCVR and QT Cards are set to issue Card Valid signals for this class of address access only. However, to save circuit card real estate and costs, not all address bits in the A24 field are decoded. Address bits A02 - A08 and A15 are decoded. Bits A09 - A14 and A16 - A23 are not decoded and if addressed, with the appropriate lower bits set, they will cause the same actions as if they were zero. The A15 bit is used to distinguish test and control commands from the more routine reading of the status register and the FIFO. This makes the definition of a "page" a little mixed up. In future versions we will decode the full address space. In practice, 16 QT Cards fill up a crate, so there is no real limitation in this somewhat sloppy usage.

The address modifier (AM) decoders on the cards are set to the following values for D32:A24:Non-Privileged VME bus transactions:

```
AM0  H  1
```

```
AM1  L  0
AM2  L  0
AM3  H  1
AM4  H  1
AM5  H  1
```

2.3 RCVR Card Commands and Data Output Format

Address [1]	Action
*0xf000 0000	Enable test mode
*0xf000 0004	Enable data taking mode
*0xf000 0008	Read status register and FIFO flags
*0xf000 000c	Shift out and read RCVR Card FIFO
*0xf000 8000	Generate single dual port fill (for QT tests)
0xf000 8004	Write test TSA and generate a FIFO write (nb, this is a write, so supply some data.)
*0xf000 8008	Enable free-running clock, generated on RCVR Card
*0xf000 800c	Clear RCVR FIFO and reset the stack

[1] The prefix 0xf000 XXXX is determined by the way the particular MBC's VME_CHIP2 is configured to map to the VME bus. The QT Crate MBCs are set for D32:A24:Non-privileged VME bus operation.

Example 1: To read the status register, the c code

```
x =  *0xf000 0008
```

will return the contents of the status register in the variable x.
x should be a long (32-bit) unsigned integer.

Example 2:
The following c code will write a test TSA to the RCVR Card:

```
*0xf000 8004 = 0xa
```

This writes the bit pattern 0000 0000 0000 1010 to the RCVR Card.

Issue a *0xf000 000c to shift out and read RCVR Card FIFO puts data out on the VME bus in the following Long Word format:

Bit	Description
31	eof
30	RCVR FIFO EF
29	RCVR FIFO HF
28	RCVR FIFO FF
27	QT FIFO EF
26	QT FIFO HF
25	QT FIFO FF
24-19	Event Type (bit-24 msb, bit-19 lsb)
18-11	Event ID (bit-18 msb, bit-11 lsb)
10	Time Stamp 102.4 microsec.
09	Time Stamp 50.2 microsec.
08	Time Stamp 25.6 microsec.
07	Time Stamp 12.8 microsec.
06	Time Stamp 6.4 microsec.
05	Time Stamp 3.2 microsec.
04	Time Stamp 1.6 microsec.
03	Time Stamp 0.8 microsec.
02	Time Stamp 0.4 microsec.
01	Time Stamp 0.2 microsec.
00	Time Stamp 0.1 microsec.

Issue a *0xf000 0008 to read status register and FIFO flags

RCVR Status Register Format

Bit	Description
31	RCVR FIFO EF
30	QT FIFO EF
29	RCVR FIFO HF
28	QT FIFO HF

```
27                  RCVR FIFO FF
26                  QT FIFO FF

25                  Data/Test flag, =1 if in Data mode, =0 if in Test mode

24                  Clock Free Run bit, should be =0 for normal operation


Issue a   0xf000 8004 to write test TSA and generate a FIFO write (nb, this
                      is a write, so you must supply some data in the format
                      shown below.)  RCVR Card must be in Test mode to use
                      this command.

Test TSA Write Data Format

(VME)Data Bit       Description

D00                 TSA 00
D01                 TSA 01
D02                 TSA 02
D03                 TSA 03
D04                 TSA 04
D05                 TSA 05
D06                 TSA 06
D07                 TSA 07
D08                 TSA 08
D09                 TSA 09
D10                 TSA 10
```

2.4 QT Card Commands

The QT Cards use A04 - A08 as "Card Select Lines" and jumper fields on the card are set to select a particular combination to set the card's address. This provides for unique addresses for 1 to 16 cards. (Actually for 1 to 31 cards, since five bits are used.) Address lines A02, A03, and A15 are used to select the particular on-card function. These are defined below. In order to save money, but not grief, it was decided to use programming discipline and beg people not to address anything that would use address lines A09, A10, A11, A12, A13, A14, or A16 - A23. (Note: The QT Crates operate in the world of D32: A24: Non-Priv. Data Access.)

```
On Card Addresses:
Address               Action
```

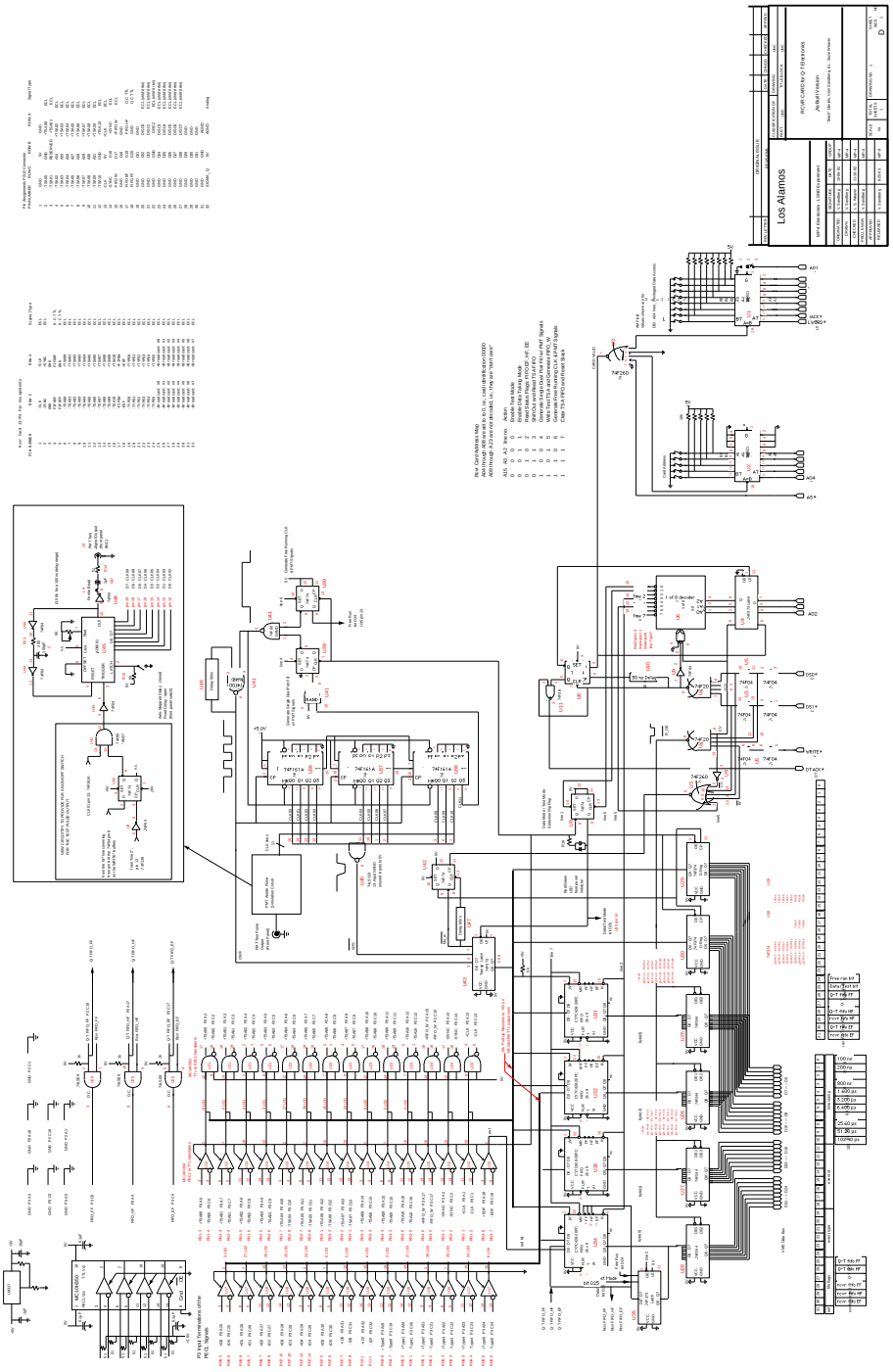

Figure 4: The LSND RCVR Card used to communicate between the QT Crate and the Trigger Broadcast Card. It distributes TSA signals and stacks the broadcast timestamps.

```
A2  A3  A15

 0   0    0          Shift out first QT Long Word, CH 0-1
 1   0    0          Shift out second QT Long Word, CH 2-3
 0   1    0          Shift our third Long Word, CH 4-5
 1   1    0          Shift out fourth Long Word, CH 6-7
 0   0    1          Clear and reset all QT FIFOs on this card

Example:  To read out the QT channels in order on card 1 you
issue the following commands:
*0xf0000010
*0xf0000014
*0xf0000018
*0xf000001c
(The way the address map is set up on the MVME167, you need to
prefix an f before these addresses so the VMECHIP2 will properly
send out a D32: A24: Non_Priv Data .)
To read out card 2:
*0x20, *0x24, *0x28, and *0x2c

and so on, and so on ...

To clear the QT FIFOs on the card issue a *0xf0008010.


-> *0xf0000008
0xf0000008: value = -385875969 = 0xe8ffffff
-> *0xf0000010
0xf0000010: value = -251989766 = 0xf0faf0fa
-> *0xf0000014
0xf0000014: value = -268832774 = 0xeff9effa
-> *0xf0000010
0xf0000010: value = -251989510 = 0xf0faf1fa
-> *0xf0000010
0xf0000010: value = -1 = 0xffffffff
-> *0xf0000010
0xf0000010: value = -1 = 0xffffffff
-> *0xf0000014
0xf0000014: value = -268832774 = 0xeff9effa
-> *0xf0000014
0xf0000014: value = -1 = 0xffffffff
-> *0xf0000018

0xf0000018: value = -252055303 = 0xf0f9f0f9
-> *0xf000001c
```

```
0xf000001c: value = -268832519 = 0xeff9f0f9
-> *0xf000001c
0xf000001c: value = -268766982 = 0xeffaf0fa
-> *0xf000001c
0xf000001c: value = -1 = 0xffffffff
->
```

2.5 Broadcast Card Introduction

The Broadcast Card provides the interface between the trigger MBC and the RCVR Cards. The Broadcast card drives the TSA cable. Its primary purpose is to broadcast the time stamp addresses selected by the trigger MBC. It receives the base TSA and the number of sequential TSAs to be broadcast from the trigger MBC. It then sends out a series of sequential TSAs along the TSA Cable. Upon transmission of the final TSA, an EOF bit is sent and the state of the trigger's binary clock is recorded in a separate FIFO in the Broadcast Card (called the diagnostic FIFO). The trigger MBC should read this FIFO to determine when the last TSA in a molecule was sent to determine if the data was stale.

As with the RCVR Card, the VME bus transactions are D32: A24: Non-Privileged Data. The Broadcast Card decodes the full A24 address space.

2.6 Broadcast Card Commands and Data Formats

Commands

Address [2]	Action
*0x0100 0100	Write a broadcast word to the broadcast input FIFO This word contains the base TSA and the number of sequential TSAs to be broadcast on the TSA Cable.
*0x0100 0110	Read the diagnostic FIFO to determine when the last TSA in a molecule was sent.
*0x0100 0108	Reset the Broadcast Card.

[2] The Trigger MBC is configured to use the prefix 0x0100 XXXX to map to VME buss addresses.

Data Formats

Broadcast Write

Bit	Description
31	Broadcast Identification 7 (msb)

30	Broadcast Identification 6
29	Broadcast Identification 5
28	Broadcast Identification 4
27	Broadcast Identification 3
26	Broadcast Identification 2
25	Broadcast Identification 1
24	Broadcast Identification 0 (lsb)
23	Event Type 5 (msb)
22	Event Type 4
21	Event Type 3
20	Event Type 2
19	Event Type 1
18	Event Type 0 (lsb)
17	not used
16	not used
15	Window size N4 (msb)
14	Window size N3
13	Window size N2
12	Window size N1
11	Window size N0 (lsb)
10	TSA 10 (msb)
09	TSA 9
08	TSA 8
07	TSA 7
06	TSA 6
05	TSA 5
04	TSA 4
03	TSA 3
02	TSA 2
01	TSA 1
00	TSA 0 (lsb)

Diagnostic Read

Bit	Description
31	Broadcast Identification 7 (msb)
30	Broadcast Identification 6
29	Broadcast Identification 5
28	Broadcast Identification 4
27	Broadcast Identification 3

```
26                Broadcast Identification 2
25                Broadcast Identification 1
24                Broadcast Identification 0 (lsb)

23                GPS Clk  80 msec. bit (note: GPS data are in BCD)
22                GPS Clk  40 msec. bit
21                GPS Clk  20 msec. bit
20                GPS Clk  10 msec. bit
19                GPS Clk   8 msec. bit
18                GPS Clk   4 msec. bit
17                GPS Clk   2 msec. bit
16                GPS Clk   1 msec. bit

15                Binary Clk 3276.8 microsec. bit
14                Binary Clk 1638.4 microsec. bit
13                Binary Clk  819.2 microsec. bit
12                Binary Clk  409.6 microsec. bit
11                Binary Clk  204.8 microsec. bit
10                Binary Clk  102.4 microsec. bit
09                Binary Clk   51.2 microsec. bit
08                Binary Clk   25.6 microsec. bit
07                Binary Clk   12.8 microsec. bit
06                Binary Clk    6.4 microsec. bit
05                Binary Clk    3.2 microsec. bit
04                Binary Clk    1.6 microsec. bit
03                Binary Clk    0.8 microsec. bit
02                Binary Clk    0.4 microsec. bit
01                Binary Clk    0.2 microsec. bit
00                Binary Clk    0.1 microsec. bit
```

2.7 Trigger-Memory Card

```
Trigger-Memory Card Programmer's Model

October 26, 2001
V. Sandberg

Listed below are the addresses and data field formats for the
trigger-memory card.  Note that the address decoder on the card
distinguishes "reads" from "writes".


Address Modifier Field (AM) decoder: Bit values
```

AM	5	4	3	2	1	0
	1	1	1	0	0	1
	H	H	H	L	L	H

A24:D32 Standard Address Non-Privledged Data Access

Address bits A23-A05 are jumper programmable (in hardware) and define the addresse space to which this card will respond. The address bits A04-A02 are decoded according to the table shown below. All VME bus transactions are D32, that is 32-bit wide "longwords". Because of this, address bits A01 and A00 are not specified. All addresses should be on longword boundaries. (Remember address values point to bytes of memory. Words are 16-bits or 2-bytes long and align on "word boundaries", which are on even byte addresses. Long words are 32-bits or 4-bytes long and longword addresses align on multiples of 4.)

Address				Action/Function
A04	A03	A02	Y	
0	0	0	0	FIFO_R read data in FIFO
0	0	1	1	Not used
0	1	0	2	STATUS_R read status register
0	1	1	3	Read out GPS time: 1ms to 20 hr bits
1	0	0	4	Read out GPS time: 1 day to 200 day + BCLK
1	0	1	5	TRIGGER_ENABLE_TOGGLE
1	1	0	6	Latch GPS time and hold it
1	1	1	7	RESET_COMMAND reset FIFO

FIFO Data Format

Dxx	Bit
00	100ns bclk
01	200ns bclk
02	400ns bclk
03	800ns bclk
04	1.6usec bclk
05	3.2usec bclk
06	6.4usec bclk
07	12.8usec bclk

08	25.6usec bclk
09	51.2usec bclk
10	102.4usec bclk
11	204.8usec bclk
12	409.6usec bclk
13	819.2usec bclk
14	1.6384msec bclk
15	3.2768msec bclk
16	Event3
17	Event2
18	Event1
19	Det5
20	Det4
21	Det3
22	Det2
23	Veto Comparator 2
24	----
25	----
26	----
27	----
28	----
29	Event4
30	Det1
31	Veto Comparator 1

GPS and Binary Clock Data Formats

Dxx	Bit	Bit
00	1ms	1day
01	2ms	2day
02	4ms	4day
03	8ms	8day
04	10ms	10day
05	20ms	20day
06	40ms	40day
07	80ms	80day

08	100ms	100day
09	200ms	200day
10	400ms	----
11	800ms	----
12	1sec	----
13	2sec	----
14	4sec	----
15	8sec	----
16	10sec	100ns
17	20sec	200ns
18	40sec	400ns
19	1min	800ns
20	2min	1.6usec
21	4min	3.2usec
22	8min	6.4usec
23	10min	12.8usec
24	20min	25.6usec
25	40min	51.2usec
26	1hr	102.4usec
27	2hr	204.8usec
28	4hr	409.6usec
29	8hr	819.2usec
30	10hr	1.6384msec
31	20hr	3.2768msec

Status Register Data Format

Dxx	Bit Name
00	QT FIFOEF
01	QT FIFOHF
02	QT FIFOFF
03	QT FFIO_HF history bit
04	QT FIFO_FF history bit
05	----
06	----
07	TRIG_ENABLE bit

```
08        Trigger Memory FIFO FF byte(0)
09        Trigger Memory FIFO FF byte(1)
10        Trigger Memory FIFO FF byte(2)
11        Trigger Memory FIFO FF byte(3)

12        ----
13        ----
14        ----
15        ----

16        Trigger Memory FIFO HF byte(0)
17        Trigger Memory FIFO HF byte(1)
18        Trigger Memory FIFO HF byte(2)
19        Trigger Memory FIFO HF byte(3)

20        ----
21        ----
22        ----
23        ----

24        Trigger Memory FIFO EF byte(0)
25        Trigger Memory FIFO EF byte(1)
26        Trigger Memory FIFO EF byte(2)
27        Trigger Memory FIFO EF byte(3)

28        ----
29        ----
30        ----
31        ----
```